

## iTOP-iMX6-驱动-PWM 控制蜂鸣器实验文档

本文档主要介绍迅为 IMX6 开发板的 PWM 配置以及控制蜂鸣器实验过程。

### 1.1 平台文件注册设备

IMX6 开发板的平台文件为：kernel\_imx/arch/arm/mach-mx6/board-mx6q\_topeet.c

进入 iTOPiMX6\_android4.4.2/kernel\_imx/arch/arm/mach-mx6/目录，使用 vi board-mx6q\_topeet.c 命令，打开 board-mx6q\_topeet.c 文件。

搜索关键词 “CONFIG\_LEDS\_CTL” ，按照 leds 模块，添加 “my\_pwm\_test” 配置，如下图所示。

```
}
#endif

/* add by neo 20171219 */
#if defined(CONFIG_PWM_TEST_CTL)
struct platform_device pwm_test_plat_device = {
    .name    = "my_pwm_test",
    .id      = -1,
};

static int itop6x_pwm_test_init(void)
{
    printk("plat: add device pwm_test\n");
    platform_device_register(&pwm_test_plat_device);
}
#endif
/*end add */

#if defined(CONFIG_BUZZER_CTL)
struct platform_device buzzer_plat_device = {
    .name    = "buzzer_ctl",
```

修改完成后，保存退出。再次打开 board-mx6q\_topeet.c 文件。

搜索关键词 “LEDS\_CTL” ,按照 leds 模块添加“ my\_pwm\_test” 配置。如下图。

```
/* add by cym 20161214 */
#ifdef CONFIG_LEDS_CTL
    itop6x_leds_init();
#endif

/* add by neo 20171218 */
#ifdef CONFIG_PWM_TEST_CTL
    itop6x_pwm_test_init();
#endif
/* end add */

#ifdef CONFIG_BUZZER_CTL
    itop6x_buzzer_init();
#endif
```

添加完成后，保存退出。

接着进入“iTOP-iMX6\_android4.4.2/kernel\_imx/drivers/char”目录，使用“vi Kconfig”命令打开 Kconfig 文件。

搜索关键词“LEDS\_CTL”，按照 leds 模块，添加“my\_pwm\_test”配置，如下图。

```
config LEDS_CTL
    bool "Enable LEDS config"
    default y
    help
        Enable LEDS config

#add by neo 20171219
config PWM_TEST_CTL
    bool "Enable PWM_TEST config"
    default y
    help
        Enable PWM_TEST config

config MAX485_CTL
    tristate "Enable MAX485 config"
    default y
    help
        Enable MAX485 config
```

修改完成后，保存退出。



```

[root@iTOP-iMX6]#
[root@iTOP-iMX6]# ls /sys/devices/platform/
ahci.0          imx-uart.1      mxc_sdc_fb.1
alarm          imx-uart.2      mxc_sdc_fb.2
anatop_reg.0    imx-wm8960.0    mxc_v4l2_capture.0
anatop_reg.1    imx2-wdt.0      mxc_v4l2_capture.1
anatop_reg.2    imx6q-ecspi.0   mxc_v4l2_output.0
anatop_reg.3    imx6q-ecspi.1   mxc_vdoa
anatop_reg.4    imx6q-flexcan.0 mxc_vpu
anatop_reg.5    imx_busfreq.0   mxs-dma-apbh
anatop_thermal.0 imx_dvfscore.0  mxs-perfmon.0
arm-pmu.0       imx_pmu.0       mxs-perfmon.1
buzzer_ctl     ion-mxc.0       mxs-perfmon.2
enet.0         leds_ctl        mxs_viim.0
fsl-ehci.0     max485_ctl      my_pwm_test
fsl-ehci.1     max8903-charger.1 oprofile-perf.0
fsl-usb2-otg   mt3326-gps      power
fsl-usb2-udc   mtk_sdio_eint.1 power.0
galcore.0      mtk_wmt.1       pwm-backlight.0
gpio-keys      mxc_asrc.0      reg-dummy
hs0038         mxc_enit.0      reg-fixed-voltage

```

至此，设备注册完成。

## 1.2 配置 PWM

### 1.2.1 board-mx6q\_topeet.h 文件修改

进入 iTOPiMX6\_android4.4.2/kernel\_imx/arch/arm/mach-mx6/目录，使用 “vi board-mx6q\_topeet.h” 命令，打开 board-mx6q\_topeet.h 文件。如下图。

```

/iTOP-iMX6_android4.4.2/kernel_imx/arch/arm/mach-mx6#
/iTOP-iMX6_android4.4.2/kernel_imx/arch/arm/mach-mx6# vi board-mx6q_topeet.h

```

注释掉 “MX6Q\_PAD\_DISP0\_DAT9\_PWM2\_PWMO,” 内容。如下图。



```

MX6Q_PAD_DISP0_DAT2__IPU1_DISP0_DAT_2,
MX6Q_PAD_DISP0_DAT3__IPU1_DISP0_DAT_3,
MX6Q_PAD_DISP0_DAT4__IPU1_DISP0_DAT_4,
MX6Q_PAD_DISP0_DAT5__IPU1_DISP0_DAT_5,
MX6Q_PAD_DISP0_DAT6__IPU1_DISP0_DAT_6,
MX6Q_PAD_DISP0_DAT7__IPU1_DISP0_DAT_7,
MX6Q_PAD_DISP0_DAT8__IPU1_DISP0_DAT_8,
/*remove by neo 20171219 */
//MX6Q_PAD_DISP0_DAT9__PWM2_PWMO,

MX6Q_PAD_DISP0_DAT10__IPU1_DISP0_DAT_10,
MX6Q_PAD_DISP0_DAT11__IPU1_DISP0_DAT_11,
MX6Q_PAD_DISP0_DAT12__IPU1_DISP0_DAT_12,
MX6Q_PAD_DISP0_DAT13__IPU1_DISP0_DAT_13,
MX6Q_PAD_DISP0_DAT14__IPU1_DISP0_DAT_14,
MX6Q_PAD_DISP0_DAT15__IPU1_DISP0_DAT_15,
MX6Q_PAD_DISP0_DAT16__IPU1_DISP0_DAT_16,
MX6Q_PAD_DISP0_DAT17__IPU1_DISP0_DAT_17,

```

在 /\* DISP\_PWM \*/后面添加如下内容。

```

/* DISP_PWM */
MX6Q_PAD_SD1_DAT3__PWM1_PWMO,          /* GPIO1[21] */

/* add by neo 20171219*/
MX6Q_PAD_SD1_DAT2__PWM2_PWMO,

/* UART1 for debug */
MX6Q_PAD_CSI0_DAT10__UART1_TXD,
MX6Q_PAD_CSI0_DAT11__UART1_RXD,

/* UART2 for BlueTooth, add by cym 20160712 */
MX6Q_PAD_EIM_D26__UART2_TXD,
MX6Q_PAD_EIM_D27__UART2_RXD,
/* end add */

```

注释掉“ MX6Q\_PAD\_SD1\_DAT2\_\_GPIO\_1\_19,” 内容。（注意，一共有两处需要注释掉）如下图。

```

MX6Q_PAD_GPIO_4__GPIO_1_4,
/* end add */

/* add by cym 20161214 */
/* remove by neo 20171219 */
//MX6Q_PAD_SD1_DAT2__GPIO_1_19,
/* end add */

static iomux_v3_cfg_t mx6q_topeet_mipi_sensor_pads[] = {
    MX6Q_PAD_GPIO_0__CCM_CLK0,          /* camera clk */

    /* remove by neo 20171218 */
    //MX6Q_PAD_SD1_DAT2__GPIO_1_19,      /* camera PWDN */

    MX6Q_PAD_SD1_CLK__GPIO_1_20,        /* camera RESET */
};

static iomux_v3_cfg_t mx6q_topeet_hdmi_ddc_pads[] = {
    MX6Q_PAD_KEY_COL3__HDMI_TX_DDC_SCL, /* HDMI DDC SCL */
    MX6Q_PAD_KEY_ROW3__HDMI_TX_DDC_SDA, /* HDMI DDC SDA */
};

```

保存，退出。

### 1.2.2 board-mx6q\_topeet.c 文件修改

进入 iTOPiMX6\_android4.4.2/kernel\_imx/arch/arm/mach-mx6/目录，使用 vi board-mx6q\_topeet.c 命令，打开 board-mx6q\_topeet.c 文件。如下图。

```
/iTOP-iMX6_android4.4.2/kernel_imx/arch/arm/mach-mx6#  
/iTOP-iMX6_android4.4.2/kernel_imx/arch/arm/mach-mx6# vi board-mx6q_topeet.c
```

注释掉 “#define TOPEET\_MIPICSI\_PWN IMX\_GPIO\_NR(1, 19)” 。并为了防止编译出错，把它设置为 NULL。如下图。

```
#define TOPEET ACCL INT IMX_GPIO_NR(1, 18)  
/* remove by leo 20171219 */  
//#define TOPEET_MIPICSI_PWN IMX_GPIO_NR(1, 19)  
#define TOPEET_MIPICSI_PWN NULL  
/* end remove */
```

因为这次实验使用的 pwm1 被红外调用，所以需要注释掉红外相关的配置。

```
#ifdef CONFIG_MX6_IR  
/* add MXC IR device */  
/* remove by leo 20171219 */  
//imx6q_add_mxc_ir(0, &mx6_topeet_ir_data);  
/* end remove */  
#endif  
  
imx6q_add_otp();  
imx6q_add_viim();  
imx6q_add_imx2_wdt(0, NULL);  
imx6q_add_dma();  
  
imx6q_add_dvfs_core(&topeet_dvfscore_data);
```

保存，退出。

### 1.2.3 make menuconfig 修改

注释掉蜂鸣器配置。

进入内核目录，依次使用命令 “cp config\_for\_iTOPiMX6\_linux .config” 、 “export ARCH=arm” 、 “make menuconfig” 。进入 Device Drivers 》 Character devices 目录，注释掉 Enable BUZZER config 配置。如下图。

[illegible]

注释掉红外配置。

进入 System Type 》 Freescale MXC Implementations 目录，注释掉 “MX6 IR Support” 选项。如下图。

```
FreeScale CPU family: (MX6-based) --->
(14) MAX_ORDER
[*] Support i.MX 6Quad Armadillo2 platform
[ ] Support i.MX 6SoloLite Armadillo2 platform
[ ] Support i.MX 6SoloLite EVK platform
[*] Support i.MX 6Quad SABRE Lite platform
[ ] Support i.MX 6Quad SABRESD platform
[*] Support i.MX 6Quad SABRE Auto platform
[*] Support i.MX 6Quad HDMIDONGLE platform
[*] Support i.MX 6Quad TOPEET platform
*** MX6 Options: ***
[*] PCI Express support
[ ]   PCI Express EP mode in the IMX6 RC/EP interconnection system
[ ]   PCI Express RC mode in the IMX6 RC/EP interconnection system
<*) USB Host 1 support
<*) FSL USB OTG support
[ ] Internal LDO in IMX6Q/DL bypass
[*] MX6 clk setting for smooth UI transtion from bootloader to kernel
[ ] MX6 IR Support
<*) i.MX Bluetooth rfkill interface support
```

保存退出。如下图。

```

x Do you wish to save your new configuration? <ESC><ESC> x
x to continue. x
x
x < Yes > < No > x

```

为防止用户原始配置信息丢失，可使用

“cp -r config\_for\_iTOPiMX6\_linux config\_for\_iTOPiMX6\_linux\_org” 命令把原始配置保存一下。（以防配置出错时，可通过 “cp -r config\_for\_iTOPiMX6\_linux\_org config\_for\_iTOPiMX6\_linux” 命令还原默认配置。）

使用 “cp -r .config config\_for\_iTOPiMX6\_linux” 命令覆盖掉对应的 linux 配置文件。

以使刚才的配置生效（在编译内核的脚本中有覆盖配置的命令）。如下图。

```
l21/iTOP-IMX6_android4.4.2/kernel_imx#  
l21/iTOP-IMX6_android4.4.2/kernel_imx# cp -r .config config_for_iTOPiMX6_linux  
l21/iTOP-IMX6_android4.4.2/kernel_imx#
```

然后使用 “./build\_gt\_kernel.sh” 命令编译内核。如下图。

```
./iTOP-IMX6 android4.4.2/kernel_imx# ./build qt kernel.sh
```

重新烧写内核到开发板。

### 1.3 驱动程序

```
#include <linux/kernel.h>
#include <linux/module.h>
#include <linux/moduleparam.h>
#include <linux/i2c.h>
#include <linux/input.h>
#include <linux/delay.h>
#include <linux/slab.h>
#include <linux/interrupt.h>
#include <linux/irq.h>
#include <linux/gpio.h>
#include <linux/platform_device.h>
#include <linux/miscdevice.h>
#include <asm/uaccess.h>
#include <linux/pwm.h>
#include <linux/fs.h>
#include <linux/io.h>
```



```
#include <linux/types.h>
#include <linux/errno.h>
#include <linux/init.h>
/*寄存器头文件*/
#include <mach/iomux-mx6q.h>
#define BUZZER_GPIO          IMX_GPIO_NR(1, 19)
#define PWM_FREQUENCY 1
struct pwm_device *my_pwm1 = NULL;
#define DEVICE_NAME          "my_pwm_test"
#define DRIVER_NAME          "my_pwm_test"
static int itop6x_pwm_test_open(struct inode *inode, struct file *file)
{
    printk(" %s  !!!\n",__FUNCTION__);
    mxc_iomux_v3_setup_pad(MX6Q_PAD_SD1_DAT2__PWM2_PWMO);
    my_pwm1 = pwm_request(1, "my_pwm1");
    if ( my_pwm1 == NULL ) {
        printk(KERN_ALERT"my_pwm1 open error.\n");
    }
    return 0;
}
static long itop6x_pwm_test_ioctl(struct file *file, unsigned int cmd, unsigned long arg)
{
    printk(" %s  !!!\n",__FUNCTION__);
    {
        if(my_pwm1 == NULL)
            return -EINVAL;
        if(arg < 0)
            return -EINVAL;
        switch (cmd) {

        case PWM_FREQUENCY:
            if(arg==0)
            {
                pwm_disable(my_pwm1);
            }
            else
            {
                pwm_config(my_pwm1, 1000000000/arg/2, 1000000000/arg);
                pwm_enable(my_pwm1);
            }
        }
    }
}
```

```
    }
    break;
    default:
        break;
    }
    return 0;
}

static int itop6x_pwm_test_release(struct inode *inode, struct file *filp)
{
    printk(" %s  !!!\n",__FUNCTION__);
    pwm_disable(my_pwm1);          // 关闭 pwm
    pwm_free(my_pwm1);             // 释放 pwm
    my_pwm1 = NULL;
    return 0;
}

static struct file_operations itop6x_pwm_test_ops = {
    .owner          = THIS_MODULE,
    .open           = itop6x_pwm_test_open,
    .release        = itop6x_pwm_test_release,
    .unlocked_ioctl = itop6x_pwm_test_ioctl,
};

static struct miscdevice itop6x_misc_dev = {
    .minor = MISC_DYNAMIC_MINOR,
    .name = DEVICE_NAME,
    .fops = &itop6x_pwm_test_ops,
};

static int itop6x_pwm_test_probe(struct platform_device *pdev)
{
    int ret;
    printk(" %s  !!!\n",__FUNCTION__);
    gpio_free(BUZZER_GPIO);
    ret = gpio_request(BUZZER_GPIO,"my_test");
    if(ret){
        printk("gpio_request BUZZER_GPIO failed!\n");
    }
    gpio_free(BUZZER_GPIO);
}
```

```
    printk("gpio_free(BUZZER_GPIO) !!\n");
    ret = misc_register(&itop6x_misc_dev);
    return 0;
}
static int itop6x_pwm_test_remove (struct platform_device *pdev)
{
    printk(" %s  !!\n",__FUNCTION__);
    misc_deregister(&itop6x_misc_dev);
    gpio_free(BUZZER_GPIO);
    return 0;
}
static int itop6x_pwm_test_suspend (struct platform_device *pdev, pm_message_t state)
{
    printk(" %s  !!\n",__FUNCTION__);
    return 0;
}
static int itop6x_pwm_test_resume (struct platform_device *pdev)
{
    printk(" %s  !!\n",__FUNCTION__);
    return 0;
}
static struct platform_driver itop6x_pwm_test_driver = {
    .probe = itop6x_pwm_test_probe,
    .remove = itop6x_pwm_test_remove,
    .suspend = itop6x_pwm_test_suspend,
    .resume = itop6x_pwm_test_resume,
    .driver = {
        .name = DRIVER_NAME,
        .owner = THIS_MODULE,
    },
};
static int itop6x_pwm_test_init(void)
{
    printk(" %s  !!\n",__FUNCTION__);
    return platform_driver_register(&itop6x_pwm_test_driver);
}
static void itop6x_pwm_test_exit(void)
{
    printk(" %s  !!\n",__FUNCTION__);
    platform_driver_unregister(&itop6x_pwm_test_driver);
}
```

```
}  
module_init(itop6x_pwm_test_init);  
module_exit(itop6x_pwm_test_exit);  
MODULE_LICENSE("Dual BSD/GPL");  
MODULE_AUTHOR("iTOPEET_neo");
```

在上面的驱动程序中，有一点需要注意的是。在“itop6x\_pwm\_test\_open”函数中我们使用了这样一句调用函数：

```
mxc_iomux_v3_setup_pad(MX6Q_PAD_SD1_DAT2__PWM2_PWMO);
```

该条命令是把 pwm 重新配置一下。因为在内核中还有很多其他的驱动程序调用了这个引脚，将其配置为别的参数。

接下来看一下“itop6x\_pwm\_test\_ioctl”函数。

```
static long itop6x_pwm_test_ioctl(struct file *filep, unsigned int cmd, unsigned long arg)  
{  
    printk(" %s  !!!\n",__FUNCTION__);  
    {  
        if(my_pwm1 == NULL)  
            return -EINVAL;  
        if(arg < 0)  
            return -EINVAL;  
        switch (cmd) {  
        case PWM_FREQUENCY:  
            if(arg==0)  
            {  
                pwm_disable(my_pwm1);  
            }  
            else  
            {  
                pwm_config(my_pwm1, 1000000000/arg/2, 1000000000/arg);  
                pwm_enable(my_pwm1);  
            }  
            break;  
            default:  
                break;  
        }  
        return 0;  
    }  
}
```



}

应用程序的 ioctl 函数给驱动函数的 “itop6x\_pwm\_test\_ioctl” 函数一共传递三个参数。第一个是 open 函数返回的句柄，即相应的设备节点。第二个参数是 cmd，为命令参数。在 “itop6x\_pwm\_test\_ioctl” 函数中通过 switch 语句来监控传过来的 cmd，并进行相应的操作。第三个参数是 arg，该参数可以是任意类型的参数，但个数只能为一。若想进行多个参数控制，可通过结构体指针来传参。其中：

pwm\_config：设置占空比。这个函数会直接操作对应 pwm 设备的寄存器。

pwm\_enable：使能 pwm 设备。同样也会写对应 pwm 设备的使能位。

在 “itop6x\_pwm\_test\_release” 函数中：

pwm\_disable：禁能 pwm 设备。同样也会写对应 pwm 设备的禁能位。

pwm\_free：所谓 free 就是让设备引用计数器减 1。对 pwm 设备而已，由于它的引用最多为 1，所以 free 其实就是让 usr\_count = 0。

## 1.4 测试过程

IMX6 开发板烧写的是最小系统。内核使用前述配置好的内核。开发板系统启动后，可以通过挂载 nfs 共享目录、tftp 传输文件和 U 盘挂载等方法，把 IM6-driver-pwm.ko 和 my\_pwm\_test 文件上传到开发板。如下图。

```
[root@iTOP-IMX6]#
[root@iTOP-IMX6]# mount -t nfs -o nolock 192.168.3.58:/home/minilinux /mnt/
[root@iTOP-IMX6]# cd /mnt/neo/pwm/user/
[root@iTOP-IMX6]# ls
IM6-driver-pwm.c      IM6-driver-pwm.o      my_pwm_test
IM6-driver-pwm.ko     Makefile              my_pwm_test.c
IM6-driver-pwm.mod.c  Module.symvers
IM6-driver-pwm.mod.o  modules.order
[root@iTOP-IMX6]#
```

使用 “insmod IM6-driver-pwm.ko” 命令加载驱动，如下图。

```
IM6-driver-pwm.mod.c  Module.symvers
IM6-driver-pwm.mod.o  modules.order
[root@iTOP-IMX6]# insmod IM6-driver-pwm.ko
itop6x_pwm_test_init  !!!
itop6x_pwm_test_probe  !!!
gpio_free(BUZZER_GPIO) !!!
[root@iTOP-IMX6]#
[root@iTOP-IMX6]#
```

使用“ ./my\_pwm\_test /dev/my\_pwm\_test on 10000” 命令运行应用程序。此时可以听见蜂鸣器的响声。按下 ctrl+c，响声停止。如下图。

```
[root@iTOP-IMX6]# ./my_pwm_test /dev/my_pwm_test on 10000
open /dev/my_pwm_test itop6x_pwm_test_open ! ! !

itop6x_pwm_test_ioctl ! ! !
PWM ON!
PWM FREQ set:10000 Hz
^C itop6x_pwm_test_release ! ! !

[root@iTOP-IMX6]#
```

使用“ ./my\_pwm\_test /dev/my\_pwm\_test on 1000” 命令运行应用程序，此时我们把高电平占用时间改为 1000ns，可以听出蜂鸣器发出响声的频率发生变化。

```
[root@iTOP-IMX6]#
[root@iTOP-IMX6]# ./my_pwm_test /dev/my_pwm_test on 1000
open /dev/my_pwm_test itop6x_pwm_test_open ! ! !

itop6x_pwm_test_ioctl ! ! !
PWM ON!
PWM FREQ set:1000 Hz
^C itop6x_pwm_test_release ! ! !

[root@iTOP-IMX6]#
```

使用“rmmod IM6-driver-pwm” 命令卸载驱动。如下图。

```
[root@iTOP-IMX6]# rmmod IM6-driver-pwm
itop6x_pwm_test_exit ! ! !
itop6x_pwm_test_remove ! ! !
[root@iTOP-IMX6]#
```

测试完毕。

## 联系方式

北京迅为电子有限公司致力于嵌入式软硬件设计，是高端开发平台以及移动设备方案提供商；基于多年的技术积累，在工控、仪表、教育、医疗、车载等领域通过 OEM/ODM 方式为客户创造价值。

iTOP-iMX6 开发板是迅为电子基于飞思卡尔最新四核处理器 iMX6 研制的一款实验开发平台，可以通过该产品评估 iMX6 处理器相关性能，并以此为基础开发出用户需要的特定产品。

本手册主要介绍 iTOP-iMX6 开发板的使用方法，旨在帮助用户快速掌握该产品的应用特点，通过对开发板进行后续软硬件开发，衍生出符合特定需求的应用系统。

如需平板电脑案支持，请访问迅为平板方案网“<http://www.topeet.com>”，我司将有能力为您提供全方位的技术服务，保证您产品设计无忧！

本手册将持续更新，并通过多种方式发布给新老用户，希望迅为电子的努力能给您的学习和开发带来帮助。

迅为电子

2017 年 11 月